

Apéndice 1

Sintaxis de Modula-2

En este apéndice se recogen de forma precisa todas las reglas sintácticas que definen el lenguaje Modula-2, utilizando la notación BNF (Backus-Naur Form). En las reglas se utilizan los siguientes *metasímbolos*.

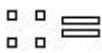




Metasímbolo	Uso
	Metasímbolo de definición. Indica que el elemento a su izquierda puede desarrollarse según el esquema de la derecha.
	Metasímbolo de alternativa. Indica que puede elegirse uno y sólo uno de los elementos separados por este metasímbolo.
	Metasímbolo de repetición. Indican que los elementos incluidos dentro de ellos se pueden repetir cero o más veces.
	Metasímbolo de opción. Indican que los elementos incluidos dentro de ellos pueden ser utilizados o no.
	Metasímbolo de agrupación. Agrupan los elementos incluidos en su interior.

Tabla A1-01: Definición y uso de los Metasímbolos.

Estos metasímbolos se escriben en el tipo de letra especial indicado para distinguirlos de los paréntesis, corchetes, etc. que forman parte del lenguaje Modula-2. También se emplearán distintos estilos de letra para distinguir los elementos simbólicos siguientes:


Estilo	Uso
<i>Elemento_no_terminal</i>	Este estilo se emplea para escribir el nombre de un elemento gramatical que habrá de ser definido por alguna regla. Cualquier elemento a la izquierda del metasímbolo  será no terminal y aparecerá con este estilo.
<i>Elemento_terminal</i>	Este estilo se emplea para representar los elementos que forman parte del lenguaje Modula-2, es decir, que constituyen el texto de un programa. Si aparecen en una regla deberán escribirse exactamente como se indican.

Tabla A1-02: Estilos de letras y uso.

La mayoría de las reglas ya han sido introducidas a lo largo del texto. Sin embargo, algunas de ellas se refieren a ciertos aspectos del lenguaje que se salen fuera de los objetivos de un primer curso de programación. Pese a todo y con la idea de ofrecer una guía de consulta rápida del lenguaje, se ha preferido realizar una definición conjunta de todas las reglas del lenguaje desde la regla más global hasta la más particular. Estas reglas, agrupadas por los grandes elementos del lenguaje, son las siguientes:

Nº	Reglas de la Unidad de Compilación
1	<i>Unidad_de_compilación ::=</i> <i>Módulo_programa</i> <i>Módulo_definición</i> <i>Módulo_implementación</i>
2	<i>Módulo_programa ::=</i> <i>Cabecera_módulo</i> <i>Bloque</i> <i>Identificador</i> .

3	<i>Módulo_definición ::=</i> <i>Cabecera_definición</i> { <i>Definición_de_módulo</i> } END <i>Identificador</i> .
4	<i>Módulo_implementación ::=</i> IMPLEMENTATION <i>Módulo_programa</i>
5	<i>Cabecera_módulo ::=</i> MODULE <i>Identificador</i> [<i>Prioridad</i>] ; { <i>Lista_importados</i> ; }
6	<i>Cabecera_definición ::=</i> DEFINITION MODULE <i>Identificador</i> ; { <i>Lista_importados</i> ; } [<i>Lista_exportados</i> ;]
7	<i>Lista_importados ::=</i> [FROM <i>Identificador</i>] IMPORT <i>Lista_de_identificadores</i>
8	<i>Lista_exportados ::=</i> EXPORT [QUALIFIED] <i>Lista_de_identificadores</i>
9	<i>Prioridad ::=</i> [<i>Expresión_constante</i>]
10	<i>Bloque ::=</i> <i>Parte_declarativa</i> <i>Parte_ejecutiva</i> END
11	<i>Definición_de_módulo ::=</i> <i>Declaración_de_constantes</i> TYPE { <i>Identificador</i> [= <i>Esquema_de_tipo</i>] ; } <i>Declaración_de_variables</i> <i>Cabecera_subprograma</i> ;
Nº	Reglas de Declaraciones
12	<i>Parte_declarativa ::=</i> { <i>Declaración</i> }
13	<i>Declaración ::=</i> <i>Declaración_de_constantes</i> <i>Declaración_de_tipos</i> <i>Declaración_de_variables</i> <i>Declaración_subprograma</i> <i>Declaración_módulo</i>
14	<i>Declaración_de_constantes ::=</i> CONST { <i>Asociación_constante</i> ; }
15	<i>Asociación_constante ::=</i> <i>Identificador</i> = <i>Expresión_constante</i>
16	<i>Declaración_de_tipos ::=</i> TYPE { <i>Definición_de_tipo</i> ; }
17	<i>Definición_de_tipo ::=</i> <i>Identificador</i> = <i>Esquema_de_tipo</i>
18	<i>Esquema_de_tipo ::=</i> <i>Tipo_simple</i> <i>Tipo_conjunto</i> <i>Tipo_formación</i> <i>Tipo_registro</i> <i>Tipo_puntero</i> <i>Tipo_subprograma</i>
19	<i>Tipo_simple ::=</i> <i>Identificador_de_tipo</i> <i>Tipo_enumerado</i> <i>Tipo_subrango</i>
20	<i>Tipo_enumerado ::=</i> (<i>Lista_de_identificadores</i>)

21	<i>Tipo_subrango ::=</i> [<i>Identificador_de_tipo</i>] [<i>Expresión_constante</i> .. <i>Expresión_constante</i>]
22	<i>Tipo_conjunto ::=</i> SET OF <i>Tipo_simple</i>
23	<i>Tipo_formación ::=</i> ARRAY <i>Tipo_simple</i> { , <i>Tipo_simple</i> } OF <i>Esquema_de_tipo</i>
24	<i>Tipo_registro ::=</i> RECORD <i>Secuencia_de_listas_de_campos</i> END
25	<i>Secuencia_de_listas_de_campos ::=</i> <i>Lista_de_campos</i> { ; <i>Lista_de_campos</i> }
26	<i>Lista_de_campos ::=</i> [<i>Lista_de_identificadores</i> : <i>Esquema_de_tipo</i> <i>Parte_variante</i>]
27	<i>Parte_variante ::=</i> CASE [<i>Identificador</i>] : <i>Identificador_de_tipo</i> OF <i>Variante</i> { <i>Variante</i> } [ELSE <i>Secuencia_de_listas_de_campos</i>] END
28	<i>Variante ::=</i> <i>Lista_de_valores</i> : <i>Secuencia_de_listas_de_campos</i>
29	<i>Tipo_puntero ::=</i> POINTER TO <i>Esquema_de_tipo</i>
30	<i>Tipo_subprograma ::=</i> PROCEDURE [<i>Lista_de_tipos_formales</i>]
31	<i>Lista_de_tipos_formales ::=</i> ([[VAR] <i>Tipo_formal</i> { , [VAR] <i>Tipo_formal</i> }]) [: <i>Identificador_de_tipo</i>]
32	<i>Declaración_de_variables ::=</i> VAR { <i>Lista_de_variables</i> ; }
33	<i>Lista_de_variables ::=</i> <i>Lista_de_identificadores</i> : <i>Esquema_de_tipo</i>
34	<i>Declaración_subprograma ::=</i> <i>Cabecera_subprograma</i> ; Bloque <i>Identificador</i> ;
35	<i>Cabecera_subprograma ::=</i> PROCEDURE <i>Identificador</i> [<i>Parámetros_formales</i>]
36	<i>Parámetros_formales ::=</i> ([<i>Grupo_de_parámetros</i> { ; <i>Grupo_de_parámetros</i> }]) [: <i>Identificador_de_tipo</i>]
37	<i>Grupo_de_parámetros ::=</i> [VAR] <i>Lista_de_Identificadores</i> : <i>Tipo_formal</i>
38	<i>Declaración_módulo ::=</i> MODULE <i>Identificador</i> [<i>Prioridad</i>] { <i>Lista_importados</i> ; } [<i>Lista_exportados</i> ;] Bloque <i>Identificador</i> ;

Nº	Reglas de Sentencias Ejecutables
39	<i>Parte_ejecutiva ::=</i> [BEGIN <i>Secuencia_de_sentencias</i>]
40	<i>Secuencia_de_sentencias ::=</i> <i>Sentencia</i> { ; <i>Sentencia</i> }
41	<i>Sentencia ::=</i> [<i>Sentencia_de_asignación</i> <i>Sentencia_de_llamada_a_procedimiento</i> <i>Sentencia_IF</i> <i>Sentencia_CASE</i> <i>Sentencia_WHILE</i> <i>Sentencia_REPEAT</i> <i>Sentencia_LOOP</i> <i>Sentencia_FOR</i> <i>Sentencia_WITH</i> <i>Sentencia_RETURN</i> <i>Sentencia_EXIT</i>]
42	<i>Sentencia_de_asignación ::=</i> <i>Variable</i> := <i>Expresión</i>
43	<i>Sentencia_de_llamada_a_procedimiento ::=</i> (<i>Identificador_de_procedimiento</i> <i>Variable</i>) [<i>Parámetros_de_llamada</i>]
44	<i>Sentencia_IF ::=</i> IF <i>Expresión</i> THEN <i>Secuencia_de_sentencias</i> { ELSIF <i>Expresión</i> THEN <i>Secuencia_de_sentencias</i> } [ELSE <i>Secuencia_de_sentencias</i>] END
45	<i>Sentencia_CASE ::=</i> CASE <i>Expresión</i> OF Caso { Caso } [ELSE <i>Secuencia_de_sentencias</i>] END
46	<i>Caso ::=</i> <i>Lista_de_valores</i> : <i>Secuencia_de_sentencias</i>
47	<i>Sentencia_WHILE ::=</i> WHILE <i>Expresión</i> DO <i>Secuencia_de_sentencias</i> END
48	<i>Sentencia_REPEAT ::=</i> REPEAT <i>Secuencia_de_sentencias</i> UNTIL <i>Expresión</i>
49	<i>Sentencia_LOOP ::=</i> LOOP <i>Secuencia_de_sentencias</i> END
50	<i>Sentencia_FOR ::=</i> FOR <i>Identificador_de_variable</i> := <i>Expresión</i> TO <i>Expresión</i> [BY <i>Expresión_constante</i>] DO <i>Secuencia_de_sentencias</i> END
51	<i>Sentencia_WITH ::=</i> WITH <i>Variable</i> DO <i>Secuencia_de_sentencias</i> END
52	<i>Sentencia_RETURN ::=</i> RETURN [<i>Expresión</i>]
53	<i>Sentencia_EXIT ::=</i> EXIT
Nº	Reglas de Expresiones
54	<i>Lista_de_valores ::=</i> <i>Valores</i> { , <i>Valores</i> }
55	<i>Valores ::=</i> <i>Expresión_constante</i> [.. <i>Expresión_constante</i>]

56	<i>Expresión_constante ::=</i> <i>Expresión_constante_simple</i> [<i>Operador_Comparador Expresión_constante_simple</i>]
57	<i>Expresión_constante_simple ::=</i> [+ -] <i>Término_constante</i> { <i>Operador_sumador Término_constante</i> }
58	<i>Término_constante ::=</i> <i>Factor_constante</i> { <i>Operador_multiplicador Factor_constante</i> }
59	<i>Factor_constante ::=</i> <i>Identificador_de_constante</i> <i>Número</i> <i>Ristra</i> <i>Conjunto</i> (<i>Expresión_constante</i>) NOT <i>Factor_constante</i> <i>Carácter_en_octal</i>
60	<i>Parámetros_de_llamada ::=</i> ([<i>Lista_de_expresiones</i>])
61	<i>Lista_de_expresiones ::=</i> <i>Expresión</i> { , <i>Expresión</i> }
62	<i>Expresión ::=</i> <i>Expresión_simple</i> [<i>Operador_comparador Expresión_simple</i>]
63	<i>Expresión_simple ::=</i> [+ -] <i>Término</i> { <i>Operador_sumador Término</i> }
64	<i>Término ::=</i> <i>Factor</i> { <i>Operador_multiplicador Factor</i> }
65	<i>Factor ::=</i> <i>Variable</i> <i>Identificador_de_procedimiento</i> <i>Identificador_de_función</i> <i>Identificador_de constante</i> <i>Número</i> <i>Ristra</i> <i>Conjunto</i> (<i>Expresión</i>) NOT <i>Factor</i> <i>Carácter_en_octal</i> <i>Llamada_a_función</i>
66	<i>Llamada_a_función ::=</i> (<i>Identificador_de_función</i> <i>Variable</i>) [<i>Parámetros_de_llamada</i>]
67	<i>Operador_comparador ::=</i> = <> # < <= > >= IN
68	<i>Operador_sumador ::=</i> + - OR
69	<i>Operador_multiplicador ::=</i> * / DIV MOD & AND
Nº Reglas de Elementos Básicos	
70	<i>Tipo_formal ::=</i> [ARRAY OF] <i>Identificador_de_tipo</i>
71	<i>Variable ::=</i> (<i>Identificador_de_variable</i> <i>Identificador_de_campo</i>) { . <i>Identificador_de_campo</i> [<i>Lista_de_expresiones</i>] ^ }
72	<i>Conjunto ::=</i> [<i>Identificador_de_tipo</i>] { <i>Lista_de_elementos</i> }
73	<i>Lista_de_elementos ::=</i> [<i>Elementos</i> { , <i>Elementos</i> }]
74	<i>Elementos ::=</i> <i>Expresión_constante</i> [.. <i>Expresión_constante</i>]
75	<i>Ristra ::=</i> " { <i>Carácter</i> } " ' { <i>Carácter</i> } '

76	<i>Número ::=</i> <i>Número_entero</i> <i>Número_real</i>
77	<i>Número_entero ::=</i> <i>dígito</i> { <i>dígito</i> } <i>dígito_octal</i> { <i>dígito_octal</i> } B <i>dígito</i> { <i>dígito_hexadecimal</i> } H
78	<i>dígito_hexadecimal ::=</i> <i>dígito</i> A B C D E F
79	<i>dígito ::=</i> <i>dígito_octal</i> 8 9
80	<i>dígito_octal ::=</i> 0 1 2 3 4 5 6 7
81	<i>Número_real ::=</i> <i>dígito</i> { <i>dígito</i> } . { <i>dígito</i> } [<i>escala</i>]
82	<i>escala ::=</i> E [+ -] <i>dígito</i> { <i>digito</i> }
83	<i>Carácter_en_octal ::=</i> <i>dígito_octal</i> { <i>dígito_octal</i> } C
84	<i>Lista_de_identificadores ::=</i> <i>Identificador</i> { , <i>Identificador</i> }
85	<i>Identificador_de_XXXX ::=</i> <i>Identificador</i> { . <i>Identificador</i> }
XXXX puede ser igual a constante, variable, tipo, campo, procedimiento o función	
86	<i>Identificador ::=</i> <i>letra</i> { <i>letra</i> <i>dígito</i> }
87	<i>letra ::=</i> A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z

Tabla A1-03: Reglas

Indice de elementos sintácticos por orden alfabético

Reglas	Nº
Asociación_constante	15
Bloque	10
Cabecera_definición	6
Cabecera_módulo	5
Cabecera_subprograma	35
Carácter_en_octal	83
Caso	46
Conjunto	72
Declaración	13
Declaración_de_constantes	14
Declaración_de_tipos	16
Declaración_de_variables	32
Declaración_módulo	38
Declaración_subprograma	34
Definición_de_módulo	11

Definición_de_tipo	17
dígito	79
dígito_hexadecimal	78
dígito_octal	80
Elementos	74
escala	82
Esquema_de_tipo	18
Expresión	62
Expresión_constante	56
Expresión_constante_simple	57
Expresión_simple	63
Factor	65
Factor_constante	59
Grupo_de_parámetros	37
Identificador	86
Identificador_de_campo	85
Identificador_de_constante	85
Identificador_de_función	85
Identificador_de_procedimiento	85
Identificador_de_tipo	85
Identificador_de_variable	85
letra	87
Lista_de_campos	26
Lista_de_elementos	73
Lista_de_expresiones	61
Lista_de_identificadores	84
Lista_de_tipos_formales	31
Lista_de_valores	54
Lista_de_variables	33
Lista_exportados	8
Lista_importados	7
Llamada_a_función	66
Módulo_definición	3
Módulo_implementación	4
Módulo_programa	2
Número	76
Número_entero	77
Número_real	81
Operador_comparador	67
Operador_multiplicador	69
Operador_sumador	68
Parámetros_de_llamada	60
Parámetros_formales	36
Parte_declarativa	12
Parte_ejecutiva	39
Parte_variante	27
Prioridad	9

Secuencia_de_listas_de_campos	25
Secuencia_de_sentencias	40
Sentencia	41
Sentencia_CASE	45
Sentencia_de_asignación	42
Sentencia_de_llamada_a_procedimiento	43
Sentencia_EXIT	53
Sentencia_FOR	50
Sentencia_IF	44
Sentencia_LOOP	49
Sentencia_REPEAT	48
Sentencia_RETURN	52
Sentencia_WHILE	47
Sentencia_WITH	51
Término	64
Término_constante	58
Tipo_conjunto	22
Tipo_enumerado	20
Tipo_formación	23
Tipo_formal	70
Tipo_puntero	29
Tipo_registro	24
Tipo_simple	19
Tipo_subprograma	30
Tipo_subrango	21
Ristra	75
Unidad_de_compilación	1
Valores	55
Variable	71
Variante	28

Tabla A1-04: Índice de elementos sintácticos por orden alfabético